

## 5                   **METHOD AND APPARATUS FOR EMULATING A PROCESSOR**

### **Field of the Invention**

10           This invention relates to electronic circuits, including but not limited to emulation of processors.

### **Background of the Invention**

15           A processor, such as, for example, a central processing unit (CPU), microprocessor, or digital signal processor among others, is the part of an electronic device that provides logic and computation functions by interpreting and executing instructions that are typically stored in memory either within the processor or memory external to the processor. Processors  
20           are integrated into many different types of devices because of their flexibility and ability to provide a computational power and logic for many different types of electronic devices, such as computers, televisions, VCRs, cellular phones, CD players, remote controls, communication switches, base stations, and numerous other electronic devices that benefit from intelligence provided  
25           by a processor.

          The processor is often considered the "heart" of a system by supplying intelligence via interaction with the circuitry with which it interfaces. This circuitry is often called peripheral circuitry. The peripheral circuitry is  
30           typically designed by the makers of different types of electronic devices. The peripheral circuitry must operate within the processor constraints, such as various busses protocols, timing, clock speed, and temperature, in order for the electronic device to be a successful product.

Because it is desirable to test the desired circuitry in an environment that provides helpful feedback to the designer, processor emulators are often utilized. Processor emulators mimic or imitate the processor's functionality, including all operations, as exactly as possible. These emulators often provide the user with the ability to stop the processor at a given clock cycle and read the processor's internal registers. In addition, emulators imitate the operational cycles of the processor. For example, when a processor utilizes a RISC (Reduced Instruction Set Computing) architecture, the processor alternates between the various operations that it provides, often leaving an operation before it is completed in order to perform another task prior to returning to the incomplete operation. In normal operation, such activity is not a problem because the speed of the processor is so fast that such interruptions are not noticeable to the user. When trying to debug a test circuit, however, such interruptions may delay the ability to test a particular operation because of the need to wait for the processor or its emulator to cycle back to a particular operation. It may not be predictable when the processor or emulator will return to a particular operation, making debugging with such a processor or its emulator very difficult. Existing emulators also do not provide the ability to select the processor operation to be tested.

20

Accordingly, there is a need for a processor emulator that efficiently tests processor operations even when the processor utilizes an interrupted or cyclic engagement of operations.

25

### Summary

A method of the present invention comprises the steps of selecting an operation that emulates at least one operation of a processor and downloading the operation into a programmable logic device. A clock speed is selected at which to operate the programmable logic device. The programmable logic device is programmed to operate at the clock speed. The operation is repeatedly tested in combination with a circuit.

30

### Brief Description of the Drawings

FIG. 1 is a block diagram of a test circuit board utilizing a processor in accordance with the invention.

5

FIG. 2 is a block diagram of a processor emulator board in accordance with the invention.

FIG. 3 is a side view of a processor emulator in accordance with the  
10 invention.

10

FIG. 4 is a block diagram of a programmable logic device in accordance with the invention.

15        FIG. 5 is a flowchart showing a method of processor emulation in accordance with the invention.

15

### Description of a Preferred Embodiment

20

The following describes an apparatus for and method of emulating and isolating processor operations such that each operation may be called at will without being interrupted by other operations that may be called in a typical processor operation cycle. One or more processor operations are emulated in a programmable logic device (PLD) that selectively mimics processor behavior by a program that downloads one or more processor operations into the PLD. Each operation may be selected and repeated continuously without interruption from any other processor operation unless it is desired for such interruption to take place. In addition, the PLD has an adjustable or variable clock speed that provides for the ability to select and change the clock speed under which the test circuit is exercised.

25

30

A block diagram of a circuit board utilizing a processor is shown in FIG. 1. A circuit board 101 has a processor socket 103 and a memory device 105, such as RAM (Random Access Memory), ROM (Read Only Memory), EPROM (Erasable Programmable Read Only Memory), and so forth. Typically, the memory device 105 stores a program run by a processor that is plugged into a processor socket 103. The circuit board 101 may also contain peripheral circuitry 107 that interfaces between the processor, memory 150, and the circuit under test, which is typically off-board. The circuit under test may be any type of electronic circuit that interfaces to a processor. The circuit board 101 also contains input/output (I/O) pins 109 through which functions performed by the system on the circuit board 101 may interact with the outside world. Data, control, and any potential clock signal may be brought to the various devices 103, 105, and 107 on the board 101. Not shown in FIG. 1, for the sake of simplicity, are traces that are routed between the various parts 103, 105, 107, and 109 on the board and generally connect these parts together. Such traces are well known in the art and vary based on the type of circuit utilized.

A block diagram of a processor emulator board is shown in FIG. 2. The emulator circuit board 201 has a special processor socket 203, programmable logic device (PLD) 205, peripherals 207, and I/O pins 209. Data and control are input via the I/O pins 209 to any necessary peripherals 207 and the PLD 205. Such data and control include the program that codes the PLD 205 to emulate one or more processor operations. The control lines are utilized to select the PLD 205 operation(s) to be tested. The PLD 205 may also emulate other functions or operations in addition to or in place of the processor operations. In addition, any data utilized to test the programmable logic may also be input and/or output through the I/O pins.

The I/O pins 201 also provide a mechanism through which a clock speed is selected and programmed into the PLD 205, i.e., the speed at which the PLD 205 or emulated processor will operate. The clock speed may be adjusted at any time during the test process. The programmable logic device 205

advantageously has the same number of I/O pins as the processor socket 203 for ease of PLD 205 and circuit board 201 design. The PLD 205 may have more pins than the processor socket 203, and it may also have fewer pins than the processor socket, although such designs would likely require  
5 additional complexity that may not be efficient.

A side view of the emulator 201 is shown in FIG. 3. The processor socket 203 is shown to go through the board and have pins that extend below the board, such that the processor socket 203 plugs into the processor socket  
10 103 of the main circuit board 101. The peripherals 207 and I/O pins 209 are not shown in FIG. 3 for the sake of simplicity.

A block diagram of an example PLD 205 is shown in FIG. 4. The PLD 205 may be, for example, a complex programmable logic device (CPLD) or a field  
15 programmable gate array (FPGA), such as CPLDs or FPGAs provided by Altera, Lattice, or Xilinx. The processor operations include any processor operation from any type of processor, including processor bus transactions. The example PLD 205 shown in FIG. 4 illustrates different types of processor operations that may be emulated in the PLD 205. In this example, a  
20 read/write quad word operation 401, a read/write double word operation 403, a read/write word operation 405, a read/write byte operation 407, a dynamic memory access (DMA) operation 409, a burst read operation 411, a burst write operation 413, a protocol stack operation 415, an interrupt process operation 417, and a protected (kernel) mode operation 419 are shown in  
25 this PLD 205.

Each operation of the PLD 205 is shown in a separate block to illustrate the isolation of each operation from the other operations. In this way, any particular processor operation may be invoked at any time and repeated as  
30 necessary until testing is complete. One or more control lines, one or more data lines, and one or more clock lines are connected to each of the PLD blocks 401 through 419. Thus, the PLD 205 may be utilized to test one or more processor operations at any speed in isolation, for example, to more

quickly track down the cause of an intermittent problem. Such capabilities are not provided by prior emulators. The PLD 205 may also be programmed one or more operations at a time, as desired. In addition, combinations of operations may be tested at one time, for example, if it is desirable to test the interface between a burst read and a burst write or a long word write after a byte write, or to test EMI (ElectroMagnetic Interference) properties of the peripheral circuitry 107. The PLD 205 may emulate any processor, such as an IBM PowerPC™, Intel Pentium III™, Sun SPARC™, Motorola MC68020™, and so forth, and any processor type, including, for example, central processing units (CPUs), microprocessors, and digital signal processors.

A flowchart showing a method of processor emulation is shown in FIG. 5. At step 501, one or more operations are selected for test. At step 503, it is determined whether the selected operation(s) is (are) presently loaded in the PLD 205. If the operation(s) is (are) not loaded in PLD 205, the operations are downloaded into the PLD at step 505, in accordance with known procedures for downloading logic and/or circuitry into a PLD 205, and the process continues with step 507. If the operation(s) is (are) loaded at step 503, the process continues with step 507. Thus, the present invention provides for selectable downloading of operations in the PLD 205, which may be useful, for example, when the PLD 205 is not large enough to emulate the entire processor or when better processor characteristics, e.g., timing, are obtained by implementing/emulating only parts of the processor at one time. Steps 503 and 505 may not be necessary when all the processor operations are downloaded or preloaded into the PLD 205.

At step 507 the processor speed is selected and programmed into the PLD 205. The processor speed may be selected based on, for example, any physical constraints that the tester may be subject to, such as the speed at which a logic analyzer operates. At step 509, the selected operation is tested by executing the code as programmed into the PLD and providing any test data to the PLD via whatever means are necessary. Test results may also be

returned via the same input/output devices. At step 511, if it is desired to adjust the speed of operation, the process continues with step 507. The speed may be adjusted up or down to accomplish a number of objectives. For example, the speed may be slowly increased until the circuit under test  
5 109 begins to malfunction, in order to help identify circuit or design flaws. If it is not desired to adjust the speed at step 511, the process continues with step 513. If at step 513, testing of the selected operation(s) is not complete, the process continues with step 509, otherwise the process continues with step 515. At step 515, if there are more operations to test, the process  
10 continues with step 501, otherwise the process ends.

Although the present invention is shown in the example where processor operations are emulated, the present invention may also be used to emulate operations and functionality other than that of a processor, such as driving  
15 signals to high or low voltage for trace continuity tests or thermal stress analysis of devices.

The present invention provides advantageous way to emulate a processor or other device without having to wait through processor operation cycles that do not provide the operation under test. Such undesirable operational  
20 cycles may affect the state of memory or internal registers of the processor and may provide undesirable results or require extra time to set to a desirable state. The present invention provides the ability to select and isolate a processor operation in order to repeatedly test that operation  
25 without interrupt from undesirable operation cycles until the desired results are obtained. Because the clock speed of the PLD may be varied according to the desires of the tester, slower operational speeds may be utilized for testing. As a result, slower, and hence less expensive, test equipment may be utilized to analyze circuit under test and the test results. This feature is  
30 also useful when very high speed processors, e.g., over 1 GHz, are utilized, and test equipment may not yet exist. The present invention provides advantage over prior emulators that do not provide the ability to adjust the speed of the processor and must follow operation cycles that prohibit the

ability to continuously test a single operation or group of operations without interruption from other operations, unless such interruption is desirable. The present invention also provides the ability to select and test any processor operation(s) that are desired. Because the PLD clock speed is adjustable, a wide range of speeds may be utilized with a test circuit in order to determine the clock speed at which a circuit will fail to operate properly. The present invention therefore provides a flexible yet efficient manner for testing devices that utilize processors or other complex circuitry without requiring expensive additional test equipment while reducing overall development time. The present invention provides for quick and efficient generation of test patterns through programming, utilizing standard VHDL (Very High speed Description Language) or VERILOG programming, of a PLD or FPGA that mimics a processor.

The present invention may be embodied in other specific forms without departing from its spirit or essential characteristics. The described embodiments are to be considered in all respects only as illustrative and not restrictive. The scope of the invention is, therefore, indicated by the appended claims rather than by the foregoing description. All changes that come within the meaning and range of equivalency of the claims are to be embraced within their scope.